removed to specific circles of players that might be a better fit or that might make playing the game more enjoyable. The generation of circles and the assigning of removed players to different circles dynamically by way of spectator voting will allow all players to continue playing and enjoying their games, albeit within different circles of players.

FIG. **3** is a simplified schematic diagram that illustrates a process for removing a player from a game based on group voting by spectators, in accordance with one embodiment. As shown in FIG. **3**, a plurality of spectators, S_1, S_2, S_3, . . . S_N, are watching a game, e.g., as set forth above with reference to FIG. **1**. During the game, spectators typically can talk to one another over an audio channel. In the case of popular games, which can attract hundreds and thousands of spectators, audio problems can occur when too many spectators try to share audio. To avoid such audio problems, the servers of the online platform sharing the game can use load balancing to divide the spectators into smaller groups. In addition, spectators can join together on their own, e.g., with friends, family, etc., to form a group that watches the game together. As shown in FIG. **3**, spectators S_1, S_2, and S_3 are part of first group **300** ("Group A"), spectators S_4 and S_5 are part of second group **302** ("Group B"), and spectators S_6, S_7, and S_8 are part of third group **304** ("Group 3").

As the spectators watch the game, the members of each group can vote to have a player removed from the game, as described above with reference to FIG. **2**. For each group, the individual member votes are transmitted to a group vote generator **306** for processing to determine the vote for the group. In one embodiment, in the course of determining the vote for the group, the vote of each member of the group is weighted based on the member's skill level in the game, as described above in more detail with reference to FIG. **2**. Thus, in the case of Group A, if spectator S_1 has a relatively high skill level in the game and spectator S_3 has a relatively low skill level in the game, group vote generator **306-1** would give more weight to spectator S_1's vote than to the spectator S_3's vote.

In one embodiment, the group vote generator **306** analyzes the weighted member votes to have a player removed from the game on a player-by-player basis and generates a single group vote for each player for which a vote was cast. By way of example, if all of the members of Group A voted to have Player 3 removed from the game, the group vote generator **306-1** would determine that the Group A vote with regard to Player 3 is to have the player removed from the game. The group vote generator **306-1** transmits the group vote with regard to each player to the input aggregator **218** (see FIG. **2**) for further processing, as will be described in more detail below.

In another example, if spectator S_4, who has a relatively high skill level in the game, votes to keep Player 2 in the game and spectator S_5, who has a relatively low skill level in the game, votes to have Player 2 removed from the game, the group vote generator **306-2** would determine that the Group B vote with regard to Player 2 is to keep the player in the game. The reason for this determination is that spectator S_4's vote would be given more weight than spectator S_5's vote because spectator S_4 has a higher skill level in the game.

With continuing reference to FIG. **3**, the group vote generators **306-1**, **306-2**, and **306-3** transmit the group votes for Group A, Group, and Group C, respectively, to input aggregator **218**. In one embodiment, the input aggregator **218** weights each group vote with regard to a player on the basis of the number of members in each group. By way of example, if the Group C vote with regard to Player 1 was to have the player removed from the game and the Group B vote with regard to Player 1 was to keep Player 1 in the game, the input aggregator would accord the Group C vote 50% more weight than the Group B vote because Group C has 3 members and Group B has 2 members. The group vote generators **306-1**, **306-2**, and **306-3** can transmit the number of members in the particular group to the input aggregator **218** at the same time the group vote with regard to a player is transmitted to the input aggregator **218** for processing.

The input aggregator **218** processes the group votes received from the group vote generators **306-1**, **306-2**, and **306-2** and transmits the group votes to crowd sourced vote tabulator **220** for further processing along the lines described above with reference to FIG. **2**. In particular, the crowd sourced vote tabulator **220** tallies the group votes received from the input aggregator **218** and displays the voting results for each player. Once the crowd sourced vote tabulator **220** has tallied the group votes and displayed the voting results, the crowd sourced vote tabulator **220** transmits the voting results to rules engine **222** for further processing to determine whether any players should be removed from the game based on the voting results. As described above, the rules engine **222** applies a set of rules to the voting results for each player to determine whether the player should be removed from the game. In one embodiment, the rules engine **222** applies a set of rules tailored specifically for group voting results. By way of example, the set of rules for group voting results can include different voting thresholds for each group to trigger automatic removal a player from the game or can restrict the players for whom each group can vote to have removed from the game, e.g., Group A can vote to remove players from team **1**, Group B can vote to remove players from team **2**, etc. If the rules engine **222** determines that the group voting results mandate that a player be removed from the game, the rules engine **222** issues a remove player command **224**. As described above with reference to FIG. **2**, the remove player command **224** is transmitted to the online game system **100** (see FIG. **1**) and instructs the online game system **100** to remove the player from the game without giving the player any opportunity to override the command.

FIGS. **4**A-**4**D show examples of player removal interfaces that enable a spectator to pay to remove a player from a game, in accordance with one embodiment. In addition to voting to remove a player from a game, spectators can also pay to remove a player from a game. FIG. **4**A shows a simplified user interface **200-1** that includes player removal interface **240**, which lists several options a spectator has to pay to remove a player from the game. As shown in FIG. **4**A, user interface **200-1** includes game view **202**, a communication channel **204**, and a player removal interface **240**. The game view **202** and communication channel can be the same as described above with reference to FIG. **2**. The player removal interface **240** lists the options a spectator has to pay to remove a player from a game. As shown in FIG. **4**A, the player removal interface **240** includes graphical buttons labeled "Pay Fixed Price," "Pay Percentage of Fixed Price," and "Bid in Auction." To pay in full to have a player removed from the game, the spectator can click on the graphical button labeled "Pay Fixed Price" to cause a player removal interface **240-1** to be displayed. Additional details regarding the player removal interface **240-1** are set forth below with reference to FIG. **4**B. To pay in part to have a player removed from the game, the spectator can click on the graphical button labeled "Pay Percentage of Fixed Price" to cause a player removal interface **240-2** to be displayed. Additional details regarding the player removal interface **240-2** are set forth below with reference to FIG. **4**C. To